POZNAN UNIVERSITY OF TECHNOLOGY

Wojciech Mioduszewski

Klasyfikacja danych opisanych za pomocą szeregów czasowych

Master's Thesis

Supervisor: dr inż. Jerzy Błaszczyński

Poznań, 2015

Contents

Chapter 1

Wstęp

Cel i zakres pracy

Cel: Opracowanie i implementacja różnych podejść do klasyfikacji danych czasowych.
Zadania:

- Zapoznać się z literaturą tematu.
- Opracować wybrane podejścia do klasyfikacji danych czasowych.
- Zaimplementować i udokumentować zaproponowan rozwiązania.
- Przeprowadzić eksperyment obliczeniowy

Początkowo celem niniejszej pracy była analiza szeregów czasowych zawierających dane ciśnienia w oku pacjentów zdrowych, oraz tych ze zdiagnozowaną jaskrą. Ponadto zamiarem było użycie do tego celu metody SAX, a następnie zbudowanie klasyfikatora potrafiącego sklasyfikować dane wytworzone przez tą metodę. Równie ważne było to, aby nie testować sposobów klasyfikacji tylko i wyłącznie na danych zebranych w celu oceny jaskry, lecz również sprawdzić jak wybrane i stworzone metody poradzą sobie w odniesieniu do innych szeregów czasowych. Kolejną rzeczą, od której należało się uniezależnić są klasyfikatory, dlatego też eksperymenty przeprowadzone zostały na kilku różnych technikach kategoryzowania instancji.

The goal and the scope of the thesis

Celem pracy jest opracowanie / wykonanie analizy / zaprojektowanie / ..... Struktura pracy jest następująca. W rozdziale 2 przedstawiono przegląd literatury na temat ..... Rozdział 3 jest poświęcony ..... (kilka zdań). Rozdział 4 zawiera ..... (kilka zdań) ..... itd. Rozdział x stanowi podsumowanie pracy.

Chapter 2

Background

2.1 Definicja szeregu czasowego

Szereg czasowy jest to seria pewnych obserwacji osadzonych w czasie. Można powiedzieć, że jest to przyporządkowanie danych liczbowych do odpowiadających im punktów w czasie, najczęściej z jednakowymi odstępami między kolejnymi wartościami.

http://www.cs.put.poznan.pl/jstefanowski/aed/TPtimeseries.pdf

2.2 Przykładowe metody analizy danych czasowych[2]

Tak obszerny problem jak analiza danych czasowych musi nieść za sobą rozmaite metody przeprowadzania tej analizy. Pomimo tego, że do ostatecznego eksperymentu wybrano tylko jedną z nich - regresję, to postanowiono przedstawić po krótce czym charakteryzują się poszczególne metody oraz ewentualnie dlaczego zostały porzucone.

2.2.1 Regresja liniowa

Regresja w odniesieniu do danych czasowych sprowadza się do estymowania liniowego trendu jaki prezentuje badany szereg. Koncepcyjnie metoda polega na stworzeniu funkcji liniowej, która w najbardziej dokładny sposób przybliża wartości na kolejnych obserwacjach. Matematyczny model regresji ma zatem następującą postać:

$$y = ax + b$$

Jako miarę błędu przyjmuje się sumę kwadratów różnicy między oszacowaniami, a wartościami właściwymi.

$$S = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

http://www.cs.put.poznan.pl/jstefanowski/aed/TPDregresjawieloraka.pdf

2.2.2 Wygładzanie ("Smoothing")

Wygładzanie to metoda starająca się zniwelować ponadprzeciętne różnice wartości między kolejnymi pomiarami. Dzięki temu podejściu można dokładniej przyjrzeć się ogólnemu zarysowi funkcji, czy jej okresowymi trendami, kosztem precyzji. Podejście to pozwala zminimalizować ewentualne szumy z badanego zbioru. Metoda ta jako argument przyjmuje szerokość okna k, w ramach którego będą uśredniane wartości. W pierwszym kroku liczy się średnią z k pierwszych wartości szeregu, następnie przesuwa okno o jeden element i znów liczy średnią z k wartości. Wyjściowy zbiór dla n-elementowego zbioru będzie miał $n - k$ wartości.

Poniżej dla porównania zaprezentowano wykres miary TFADJ w przeciągu doby 2.1, oraz jego wygładzony odpowiednik 2.2. Jak widać operacja wygładzania zredukowała skrajne wychylenia szeregu i wyklarowała paraboliczną tendencję tego zbioru.

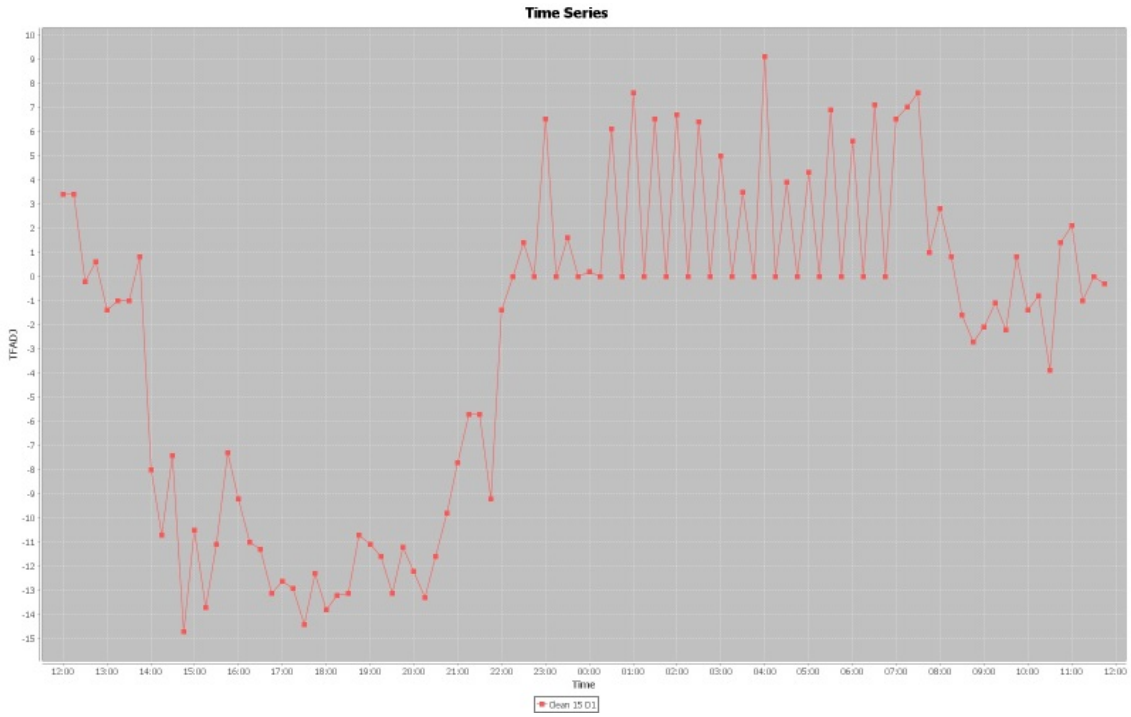

Figure 2.1: Czysty sygnał TFADJ

2.2.3 Modele ARIMA

Model ARIMA (ang. Autoregressive Integrated Moving Average) koncepcyjnie składają się z trzech części - jest to autoregresja, integracja oraz średnia ruchoma.

Autoregresja jest to idea, która skupia się na wyrażeniu bieżącej wartości na podstawie$n$ poprzednich. Dla przykładu, funkcja dla $n = 2$ wygląda w ten sposób:

$$x_t = \omega_{t-2}x_{t-2} - \omega_{t-1}x_{t-1} + \omega_t$$

Średnia ruchoma ideowo jest bardzo zbliżona do autoregresji, jednak skupia się na zaburzeniach (ang. lags) w szeregu, a nie bezpośrednio na wartościach. Wzór przedstawia

Figure 2.2: Wygładzony sygnał TFADJ

się analogicznie jak w autoregresji.

Integracja z kolei pozwala na zastosowanie modelu ARIMA do procesów niestacjonarnych, które da się sprowadzić do procesów stacjonarnych dzięki przekształceniu oryginalnego sygnału na różnice (ang. difference equations) pomiędzy wartością obecną, a poprzednią.

Podsumowując, modele ARIMA doskonale spisują się w prognozowaniu wartości, bazując na danych historycznych szeregu, jednak nie są najlepszym wyborem jako metoda klasyfikacji. Ponadto metoda nie jest najłatwiejsza matematycznie. Biorąc pod uwagę te argumenty, metoda ta została odrzucona.

2.2.4 Analiza spektralna (widmowa)

Celem analizy spektralnej (ang. spectral analysis) szeregów czasowych jest zidentyfikowanie najczęściej powtarzających się wzorców w czasie, a następnie przybliżenie danego szeregu do procesu okresowego określonego wzorem:

$$x_t = U_1 \cos(2\Pi\omega t) + U_2 \sin(2\Pi\omega t)$$

gdzie amplituda funkcji to $A = \sqrt{U_1^2 + U_2^2}$.

Jak widać jest to kolejn metoda, za którą stoi solidne matematyczne zaplecze i która wymierzona jest w szeregi cykliczne, z którymi w zbiorze danych pacjentów jaskry nie mamy do czynienia, zatem metodę tę odrzucono.

2.3 Badane zbiory danych

Jak wspomniano wcześniej, podstawowy zbiór, który miał zostać zbadany to pacjenci z jaskrą (lub bez). Z tego powodu będzie on opisany nieco szerzej niż pozostałe zbiory

2.3.1 Pacjenci

Zbiór dostarczony przez dr. J. Błaszczyńskiego. Zbiór zawiera pomiary dla 116 pacjentów, z których 65 to pacjenci zdrowi, a 51 to pacjenci ze zdiagnozowaną jaskrą. Dla każdego pacjenta zebrano 288 pomiarów (pomiar co 5 minut przez całą dobę) tzw. TFADJ <tu rozwinięcie?> będącym przekształceniem zmierzonego w tym momencie ciśnienia w oku. Poniżej poglądowe zdjęcie metody zbierania pomiarów:

Figure 2.3: Metoda zbierania pomiarów od pacjentów

2.3.2 Dane prof. Eamonna Keogh'a[1]

Zbiory pobrane od prof. Eamonn'a nie zawierają opisu poszczególnych zbiorów, dlatego też ich tu nie przedstawiono. Jednak przetwarzając dane można było wyciągnąć z nich pewne interesujące statystyki, które przedstawiono w poniższej tabeli (w ostatnim wierszu porównawczo zestawiono zbiór dr. Błaszczyńskiego):

Table 2.1: Szczegóły danych

nazwa zbióru	**liczba instancji**	**liczba klas**	**podział na klasy**	**długość szeregu**
ECG200	200	2	'-1' - 67 rekordów '1' - 133 rekordów	96
ECGFiveDays	884	2	'1' - 442 rekordów '2' - 442 rekordów	136
TwoLeadECG	1162	2	'2' - 581 rekordów '1' - 581 rekordów	82
Yoga	3300	2	'1' - 1530 rekordów '2' - 1770 rekordów	426
MoteStrain	1272	2	'2' - 587 rekordów '1' - 685 rekordów	84
ItalyPowerDemand	1096	2	'1' - 547 rekordów '2' - 549 rekordów	24
ChlorineConcentration	4307	3	'1' - 1000 rekordów '3' - 2307 rekordów '2' - 1000 rekordów	166
Two Patterns	5000	3	'2' - 1248 rekordów '3' - 1245 rekordów '4' - 1201 rekordów '1' - 1306 rekordów	128
Wafer	7174	2	'1' - 6402 rekordów '-1' - 762 rekordów	152
InlineSkate	650	7	'2' - 100 rekordów '3' - 103 rekordów '7' - 62 rekordów '6' - 98 rekordów '4' - 108 rekordów '5' - 117 rekordów '1' - 62 rekordów	1882
Pacjenci	*116*	*2*	*'0' - 65 rekordów* *'1' - 51 rekordów*	*288*

2.4 Special characters

1. Non-breaking space can be inserted using Ctrl-space. It produces "~" in LaTeX code.
2. A normal, inter-word space can be inserted using Ctrl-Alt-space. It produces "\ " in LaTeX code. This type of space is useful for formatting spacing after dots, e.g. here. By default LaTeX produces here a longer space used for separating whole sentences.
3. A thin space can be produced by Ctrl-Shift-space, e.g.here. It produces "\," in LaTeX code.
4. Sentence-ending space can be inserted using Ctrl-., which produces "\@." in LaTeX code. This type of space is useful in sentences ending with a capital letter. In such cases LaTeX recognizes the last word as a acronym and places a regular inter-word space instead of inter-sentence space. Consider the following example:

   This can be achieved by using HTTP. This protocol...

5. Hyphenation indicator can be inserted using Ctrl- –, which is used for marking possible places of hyphenation, e.g. democracy.

2.5 Figures

The figures should be put in floats, like Fig. 2.4. You can also reference figures using prettyref package like this: Fig. 2.4.

Figure 2.4: Example figure

It is possible to combine several pictures inside one float. Just insert a float inside a float. See Fig. 2.5 for example. Please note the horizontal spacing between subfigures.

2.6 Tables

Tables should have captions above like Table 2.2. Use small sans-serif fonts inside tables.

(a) The first subfigure

(b) The second subfigure

Figure 2.5: Example figure

Table 2.2: Example table

Column 1	Column 2	Column 3
One	1	4
Two	2	5
Three	3	6

2.7 Source code examples

There are a few different methods of including sample codes:

1. Using standard LyX-Code style:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

   Note 1: Empty lines must contain at least one single space to remain visible.

   Note 2: There is no way to activate automatic syntax highlighting inside LyX-Code. However, you can use normal inline formatting inside.

   Note 3: Lyx-Code can contain special characters, so it can be used to produce some ASCII art, e.g.:

```
Test 1 ──╫ Test 2
```

2. By inserting *Program Listing*:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
```

```
    return 0;
}
```

Note: By default the `lstlisting` environment does not add any left margin. You can change it by adding `xleftmargin` in the *Settings* ▷ *Advanced* dialog box, e.g.:

```
    procedure sayHello()
```

3. By inserting LaTeX Code (ERT block) and using `codeblock` environment:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

4. The listings package can produce floats by itself. See Listing. 2.1 for example.
5. And finally, You can include code from external file:

```
\documentclass[11pt,a4paper,polish,thesis]{dcsbook}

\usepackage[utf8]{inputenc}
\usepackage{babel}
\setcounter{secnumdepth}{4}
\setcounter{tocdepth}{3}

\begin{document}
```

2.8 Math

Can be put inline like this: $S = \sum_{i=1}^{i=K} x_i^2$ or in dedicated lines:

$$S = \sum_{i=1}^{i=K} x_i^2$$

Listing 2.1: The Hello World program in C

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

The equations can be also numbered like equation 2.1.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2.1}$$

2.9 Algorithms

Use dcsalg package or directly algorithmicx package.

Chapter 3

Concept and Design of the System

Chapter 4

Implementation

Chapter 5

Performance Evaluation

Chapter 6

Conclusions

Appendix A

Users Guide

Bibliography

[1] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/, ostatni dostep 24.09.2015.

[2] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2010.